Code:-

```
#include <stdio.h>

#include <stdlib.h>


typedef struct tree{

int data;

struct tree *lchild;

struct tree *rchild;

}tree;


tree* insert(tree *root,int num){

  tree *new_node=(tree*)malloc(sizeof(tree));

  new_node->data=num;

  new_node->lchild=new_node->rchild=NULL;

  if(root==NULL){

    root=new_node;

    return root;

  }

  tree *temp=root;

  while(temp!=NULL){

    if(num<temp->data){

      if(temp->lchild==NULL){

        temp->lchild=new_node;

        break;

      }

      else{

        temp=temp->lchild;
```

```
      }
    }
    else{
      if(temp->rchild==NULL){
        temp->rchild=new_node;
        break;
      }
      else{
        temp=temp->rchild;
      }
    }
  }
  return root;
}

void in_order(tree *root){
  if(root!=NULL){
    in_order(root->lchild);
    printf("%d\t",root->data);
    in_order(root->rchild);
  }
}

void pre_order(tree *root){
  if(root!=NULL){
    printf("%d\t",root->data);
    in_order(root->lchild);
```

```
    in_order(root->rchild);
  }
}
void post_order(tree *root){
  if(root!=NULL){
    in_order(root->lchild);
    in_order(root->rchild);
    printf("%d\t",root->data);
  }
}

tree* create(tree *root){
  int n,num;
  printf("How many elements You want to Add : ");
  scanf("%d",&n);
  for(int i=0;i<n;i++){
    printf("Enter %d element : ",i+1);
    scanf("%d",&num);
    root=insert(root,num);
  }
  return root;
}

void search(tree *root,int key){
  if(root!=NULL){
    if(key==root->data){
      printf("Key found\n");
```

```
      return;
      }
    else if(key<root->data){
      search(root->lchild,key);
    }
    else if(key>root->data){
      search(root->rchild,key);
    }
  }
  else{
      printf("Key not Found\n");
    }
}

tree* find(tree* temp)
{
    tree* p = temp;
  if(p->lchild!=NULL){
    while (p && p->lchild != NULL)
        p = p->lchild;
  }
  else{
    while (p && p->rchild != NULL)
        p = p->rchild;
  }
```

```
    return p;
}

tree* delete(tree* root, int key)
{
    if (root == NULL)
        return root;
    if (key < root->data)
        root->lchild = delete(root->lchild, key);

    else if (key > root->data)
        root->rchild = delete(root->rchild, key);

    else {
        if (root->lchild == NULL) {
            tree* temp = root->rchild;
            free(root);
            return temp;
        }
        else if (root->rchild == NULL) {
            tree* temp = root->lchild;
            free(root);
            return temp;
        }
        tree* temp = find(root->rchild);
        root->data = temp->data;
        root->rchild = delete(root->rchild, temp->data);
```

```
    }
    return root;
}

int main(void) {
  int c,key;
  tree *root=NULL;
  do{
    printf("\nEnter Your Choice: \n1.Insert\n2.Display\n3.Display Level
Wise\n4.Search\n5.Delete\n6.Mirror image\n7.Exit : ");
    scanf("%d",&c);
    switch(c){
      case 1:
      root=create(root);
      break;

      case 2:
        printf("In Order:\t");
        in_order(root);
        printf("\nPre Order:\t");
        pre_order(root);
        printf("\nPost Order:\t");
        post_order(root);
        printf("\n");
      break;
      case 4:
        printf("Enter Element : ");
        scanf("%d",&key);
```

```
      search(root,key);
     break;
     case 5:
      printf("Enter Element You want to Delete: ");
      scanf("%d",&key);
      root=delete(root,key);
     break;
   }
  }while(c!=7);
  return 0;
}
```

Output:-

```
> make -s
> ./main

Enter Your Choice:
1.Insert
2.Display
3.Display Level Wise
4.Search
5.Delete
6.Mirror image
7.Exit : 1
How many elements You want to Add : 9
Enter 1 element : 38
Enter 2 element : 14
Enter 3 element : 8
Enter 4 element : 23
Enter 5 element : 18
Enter 6 element : 15
Enter 7 element : 56
Enter 8 element : 45
Enter 9 element : 70
```

```
Enter Your Choice:
1.Insert
2.Display
3.Display Level Wise
4.Search
5.Delete
6.Mirror image
7.Exit : 2
In Order:   8   14  15  18  23  38  45  56  70
Pre Order:  38  8   14  15  18  23  45  56  70
Post Order: 8   14  15  18  23  45  56  70  38

Enter Your Choice:
1.Insert
2.Display
3.Display Level Wise
4.Search
5.Delete
6.Mirror image
7.Exit : 4
Enter Element : 18
Key found
```

```
Enter Your Choice:
1.Insert
2.Display
3.Display Level Wise
4.Search
5.Delete
6.Mirror image
7.Exit : 5
Enter Element You want to Delete: 38

Enter Your Choice:
1.Insert
2.Display
3.Display Level Wise
4.Search
5.Delete
6.Mirror image
7.Exit : 2
In Order:   8   14  15  18  23  45  56  70
Pre Order:  45  8   14  15  18  23  56  70
Post Order: 8   14  15  18  23  56  70  45
```